UCDCS at COLIEE 2026: A Multi-Stage Framework for Legal Case Retrieval via Structural Abstraction and Specialised LLMs

Yuchen Zhang
School of Computer Science
University College Dublin
Dublin, Ireland
yuchen.zhang3@ucdconnect.ie

David Lillis
School of Computer Science
University College Dublin
Dublin, Ireland
david.lillis@ucd.ie

Abstract

Automated legal case retrieval is a critical yet challenging task due to the extreme length of judicial documents and the complexity of judicial reasoning. In this paper, we present our multi-stage framework for the COLIEE 2026 Task 1 (Legal Case Retrieval). To address the challenges of long-form legal texts, we first employ a structural abstraction strategy that distills cases into key factual and logical components. Our retrieval pipeline utilises a hybrid strategy combining BM25 with BGE-M3 dense embeddings, establishing a high-recall foundation (≈0.85). For the subsequent re-ranking stage, we move beyond general-purpose semantic matching by leveraging fine-tuned MonoT5 models and SaulLM-7B, a specialised legal large language model. This transition allows the system to prioritise logic over surface-level topical similarity. Among the three runs we submitted, the best performance achieved by the proposed framework reached a final precision of 0.2480 and an F1-score of 0.2645 on the evaluation set, improving upon the retrieval-only baselines. These results indicate that combining hybrid retrieval with domain-adapted re-ranking is a promising approach.

CCS Concepts

• **Information systems** → **Retrieval models and ranking**; • **Applied computing** → **Law**.

Keywords

Legal Case Retrieval, Large Language Models, Structural Abstraction, Judicial Reasoning

ACM Reference Format:
Yuchen Zhang and David Lillis. 2026. UCDCS at COLIEE 2026: A Multi-Stage Framework for Legal Case Retrieval via Structural Abstraction and Specialised LLMs. In *COLIEE 2026, June 12, 2026, Singapore.* ACM, New York, NY, USA, 8 pages.

1 Introduction

As the body of case law continues to expand, it becomes increasingly challenging for legal professionals to identify relevant precedents efficiently. Automated legal Information Retrieval systems can help address this difficulty by retrieving prior cases that may support judicial decision-making. The Competition on Legal Information Extraction and Entailment (COLIEE) provides a standardised benchmark for evaluating such systems, with a particular focus on legal retrieval and entailment tasks [12].

In this paper, we focus on Task 1: Legal Case Retrieval, which aims to automatically retrieve "noticed cases" for a given query case. Following the official task definition[1], a noticed case refers to a prior case that is referenced by the query case and serves as supporting evidence for its legal decision. To ensure a realistic evaluation setting, all citation information is removed from the query cases, requiring systems to infer these relationships solely based on textual and semantic similarity. Given a corpus of Federal Court of Canada case law, the task requires fully automatic retrieval without any human intervention, and participants must predict the noticed cases without access to the test labels before submission.

Legal case retrieval poses several challenges. First, legal documents are often lengthy and structurally complex, making it difficult for traditional keyword-based methods to capture deep semantic relationships. Second, legal relevance is not determined by lexical overlap alone, and accurate retrieval often depends on semantic and reasoning cues beyond exact term matching [9]. Finally, in the COLIEE Task 1 setting, the number of relevant cases varies substantially across queries, which complicates threshold selection and result filtering.

To address these challenges, we propose a multi-stage retrieval and re-ranking framework. Our approach first leverages large language models (LLMs) to generate structured summaries that capture key legal elements of each case. We then employ a hybrid retrieval strategy that combines lexical matching using BM25 with semantic similarity based on BGE-M3 embeddings to generate candidate cases [3]. Finally, we apply domain-specific re-ranking to refine the retrieved candidates and improve final retrieval performance. This design is intended to better capture semantic relationships between legal cases and produce more accurate retrieval results.

Our main contributions are as follows:

- We developed an LLM-based legal abstraction step to generate structured case summaries, reducing noise while preserving essential legal information;
- We build a hybrid retrieval framework that combines lexical matching with dense semantic representations for legal case retrieval over long documents;
- We conduct a comparative study of general-purpose and domain-specific re-ranking models, demonstrating the effectiveness of legal-specialised models in capturing fine-grained case relationships.

[1] https://coliee.org/COLIEE2026/tasks/task1

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
COLIEE 2026, Singapore
© 2026 Copyright held by the owner/author(s).

2 Task Description

Task 1 of COLIEE focuses on legal case retrieval over a corpus of Federal Court of Canada decisions. The dataset consists of a collection of case law documents, where each case may serve as a query or a candidate case.

The training data used in this study is constructed from the union of the previous year's Task 1 training and test collections, yielding a candidate pool of 7,709 cases. The test collection consists of 1,848 candidate cases. Only the cases appearing as keys in the JSON file are treated as query cases; the system retrieves relevant noticed cases for each query from the corresponding candidate pool. In the training set, the annotations are provided in JSON format, where each query case is mapped to a set of ground-truth noticed case identifiers. For example:

```
{
 "000001.txt":
    ["000005.txt", "012101.txt"],
 "003423.txt":
    ["398421.txt", "012101.txt", "173651.txt"],
 "012831.txt":
    ["000001.txt"],
 ...
}
```

For the test set, only the query cases are provided, and the system is required to retrieve the corresponding noticed cases from the entire corpus. The retrieval process must be fully automatic and cannot rely on any manual intervention.

The task is formulated as a retrieval problem, where systems are expected to return a ranked list of candidate cases for each query, ordered by relevance.

System performance is evaluated using the standard Information Retrieval metrics of Precision, Recall, and F1-score, which together measure the accuracy and completeness of the retrieved results.

3 Related Work

Legal Information Retrieval, and Legal Case Retrieval in particular, has evolved alongside broader developments in Natural Language Processing, moving from keyword-based retrieval to semantic and neural methods. In the COLIEE competition, prior systems have mainly addressed three recurring challenges: vocabulary mismatch, the extreme length of legal documents, and the need to capture complex legal reasoning [11].

3.1 Lexical and Hybrid Retrieval

Traditional lexical methods such as BM25 [20] remain strong baselines because they perform well when relevant documents share explicit terms with the query. However, their reliance on surface-level term matching makes them vulnerable to vocabulary mismatch. To mitigate this limitation, many recent systems combine lexical retrieval with dense semantic retrieval [9, 17]. Transformer-based encoders such as BERT and its variants [8, 16], as well as Sentence-BERT [19], have made it possible to model semantic similarity beyond exact word overlap. One recent COLIEE submission reported that combining BM25 with dense retrievers can improve candidate recall [17].

3.2 Neural Re-ranking and Domain Adaptation

Neural re-ranking plays a crucial role in modern retrieval systems because it allows more fine-grained modelling of query-document interactions [27]. MonoT5 [18] has shown strong performance in re-ranking tasks by reformulating ranking as sequence generation. However, the domain gap between general-purpose training data and legal language remains a challenge [2, 9]. Domain adaptation techniques, such as fine-tuning on legal corpora or using domain-specific models like LEGAL-BERT [2] can improve performance in Legal NLP tasks. A prior COLIEE system has also explored related strategies, including LLM-based summarization and fine-tuned ranking models [24]. In this paper, we extend this line of work by comparing general-purpose and domain-adapted re-ranking models in our retrieval pipeline.

3.3 Legal-Specific Large Language Models

LLMs have also opened new possibilities for legal text modelling in the last two years, especially for tasks that require reasoning over facts, legal principles, and case relationships. Techniques such as Chain-of-Thought prompting [23] and Retrieval-Augmented Generation [13] have shown strong potential. In the legal domain, specialised models trained on legal corpora, such as SaulLM-7B [4], provide improved understanding of legal terminology and reasoning, making them suitable for identifying relationships between cases.

3.4 Summarisation and Information Distillation

The extreme length of legal documents presents significant challenges for neural models, often leading to information loss in long-context processing. This has resulted in a variety of approaches to address this issue, including representing long legal documents as alternative data structures such as graphs [26]. Other structure-aware approaches that leverage domain knowledge include LSDK-LegalSum [10], which emphasises partitioning judicial judgments into functional logical sections. In this work, we draw inspiration from the UMNLP team's strategy at COLIEE 2024 [6], which demonstrated the effectiveness of distilling cases into concise legal *propositions* to fit within model constraints while preserving essential facts. These studies support the broader idea that legal information distillation can improve retrieval effectiveness under long-context constraints.

4 Methodology

As shown in Figure 1, we propose a multi-stage retrieval and re-ranking framework designed to address the challenges of long-context legal documents, implicit semantic relationships, and the absence of explicit citations in COLIEE Task 1. Given a query case q and a corpus C, the system aims to retrieve a ranked list of noticed cases. The overall pipeline consists of four stages: (1) legal semantic abstraction, (2) hybrid first-stage retrieval, (3) neural re-ranking, and (4) result selection and filtering. These stages are described in the following sections.

4.1 Legal Semantic Abstraction

Legal case documents are often lengthy and complex, and not all parts of a full judgment are equally useful for notice-case retrieval.

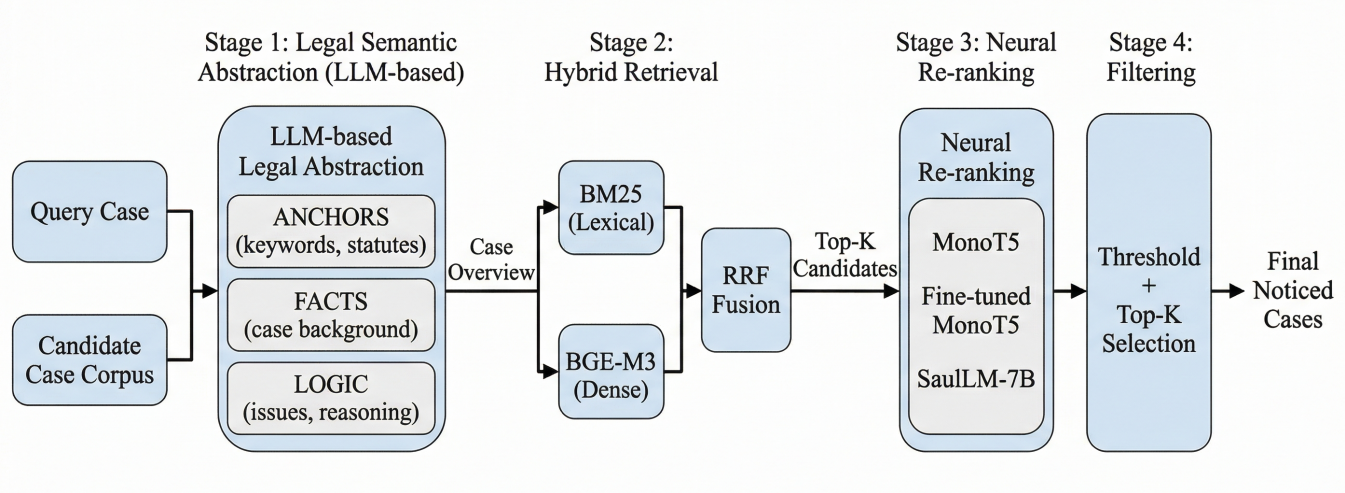

Figure 1: Overview of the proposed multi-stage legal case retrieval framework.

In particular, retrieval is more likely to benefit from information about the core legal issue, the reasoning of the decision, and the most salient facts than from procedural background or other material that is peripheral to the dispute [7, 21]. To reduce this noise, we generate a structured case overview for each document using the `deepseek-chat` LLM through an API-based prompting pipeline. The overview generation step is designed to preserve legally salient information while compressing the original judgment into a shorter and more retrieval-oriented representation.

Each case is converted into a structured summary with three predefined sections:

- **[ANCHORS]**: key legal terms and cited statutory provisions;
- **[LOGIC]**: the main legal issue and the core reasoning of the decision;
- **[FACTS]**: a concise description of the factual background.

The prompt explicitly instructs the model to extract legal keywords, identify exact statute names and section numbers, summarise the core legal issue and *ratio decidendi*, and provide a brief factual summary. It also includes a task-specific constraint requiring the tag *Foreign Law Interpretation* when the case turns on the interpretation of foreign law. To ensure consistent formatting, the model is required to output the exact section headers and terminate the response with a special `[END]` marker, after which the output is post-processed and cleaned. The full prompt is provided in Appendix A.

The intent of this structured abstraction is to reduce the length of the document while preserving information that is more likely to support notice-case retrieval. The resulting overviews are then used as inputs for both first-stage retrieval and downstream re-ranking.

4.2 Hybrid First-stage Retrieval

To obtain a high-recall candidate pool, we adopt a hybrid retrieval framework that combines lexical matching with dense semantic retrieval. All retrieval operations are performed on the generated case overviews.

Lexical Retrieval. We use BM25 [20] as a strong baseline for term-based matching. Documents are preprocessed using tokenisation and stemming. We tuned the BM25 hyperparameters empirically on COLIEE 2026 training set and found that $k_1 = 1.6$ and $b = 0.9$ gave the best performance for our legal retrieval task.

Dense Retrieval. We employ BGE-M3 [3], a dense embedding model capable of encoding long textual inputs. Each case overview is mapped to a dense vector, and similarity is computed using inner product.

Fusion. The ranked lists from BM25 and dense retrieval are combined using Reciprocal Rank Fusion (RRF) [5], with a smoothing parameter $K = 60$, following the original paper. RRF was chosen on the basis that it is a commonly-used fusion method that is straightforward to implement [14]. The fused score $RRF(d)$ for a document d is defined as:

$$\mathrm{RRF}(d) = \sum_{r \in R} \frac{1}{K + \mathrm{rank}_r(d)}$$

where R denotes the set of retrieval methods and $\mathrm{rank}_r(d)$ is the rank of document d in the set of results returned by method r. We retain the top 200 fused candidates for the next stage. We chose this cut-off to match the number of results returned by both BM25 and dense retrieval, so that fusion preserves the full candidate set contributed by each retriever while still providing a manageable pool with good recall for downstream re-ranking.

4.3 Neural Re-ranking

To further refine the candidate set, we apply neural re-ranking models that capture fine-grained relationships between query and candidate cases. Each query-candidate pair is formatted as a text pair and scored independently. We submitted three final runs to COLIEE, each corresponding to a different re-ranking pipeline: Pipeline-Base (submitted with the label `ucdcs1`), Pipeline-FineTune (`ucdcs2`), and Pipeline-SaulLM (`ucdcs3`). All three pipelines share the same document abstraction and first-stage retrieval framework,

and differ only in the re-ranking model used in the second stage. The following paragraphs describe the three submitted pipelines.

Pipeline-Base (`ucdcs1`). We use MonoT5 [18], a sequence-to-sequence model trained on MS MARCO [1], as a general-purpose re-ranker. The model takes a query-case pair as input and produces a relevance score in a zero-shot setting.

Pipeline-FineTune (`ucdcs2`). To reduce the domain gap between general-domain ranking data and legal case retrieval, we fine-tune MonoT5 on the training set, starting from `castorini/monot5-base-msmarco`. The task is formulated as binary sequence generation over query-document pairs. Both queries and candidate documents are represented by the generated case overviews rather than the full judgments.

Positive pairs are constructed from gold noticed-case labels. Negative pairs are limited to at most five per query. We define *hard negatives* as non-relevant candidate documents retrieved for the same query by the first-stage candidate pool. After excluding the query document itself, gold positives, and documents without available overviews, we take the top 30 such non-relevant candidates as a hard-negative pool and randomly sample up to five from this pool. Random negatives are used only as a fallback when fewer than five hard negatives are available, for example if the candidate pool is missing or if too few valid non-relevant candidates remain after filtering. To avoid query leakage, the data are split at the query level into training and development sets with an 80/20 split. We use a maximum input length of 1024 tokens to preserve richer legal context, and train the model for 2 epochs with a learning rate of 1×10^{-5}.

Pipeline-SaulLM (`ucdcs3`). Our third run is based on a domain-specific large language model, SaulLM-7B [4], which is pre-trained on large-scale legal corpora. We formulate the re-ranking task as an instruction-following problem, prompting the model to determine whether a candidate case provides decision-supporting evidence for a query case.

Instead of directly generating textual outputs, we convert the model's predictions into continuous relevance scores via a logit-based scoring mechanism. During inference, we extract the raw logits corresponding to the first generated token. Let L_{Yes} and L_{No} denote the logits for the tokens "Yes" and "No", respectively. We then define the relevance score S for a query–document pair as:

$$S = \frac{e^{L_{\text{Yes}}}}{e^{L_{\text{Yes}}} + e^{L_{\text{No}}}} \tag{1}$$

This formulation provides a relative relevance score for ranking, enabling fine-grained ranking and more stable threshold-based filtering in the final stage. In practice, we use a context window of up to 4,096 tokens to accommodate the structured case overviews.

4.4 Result Selection and Filtering

A key challenge in Task 1 is that the number of noticed cases varies substantially across queries and is unknown at inference time. We therefore require a decision rule that converts the continuous scores produced by the neural re-rankers into a discrete set of predictions. To address this issue, we apply an adaptive filtering procedure whose parameters are tuned on a hold-out development split of the training data and then fixed for inference on the unseen test set. The goal is to balance precision and recall while accounting for query-level variation in the number of relevant cases.

Hybrid Cutoff Strategy. Our main method is a Hybrid Cutoff strategy that combines a rank-based limit with a score threshold. For each query, we first retain the top $K = 7$ candidates according to the re-ranking scores. We then apply a threshold $\tau = 0.75$ and remove candidates whose scores fall below this value. This two-stage procedure restricts the output to highly ranked candidates while filtering out low-scoring ones, which helps maintain precision. However, because the final output is selected from a much smaller subset of the top-200 candidate pool, relevant cases that are not assigned sufficiently high re-ranking scores may fall below the final cut-off and therefore be excluded from the submitted predictions.

Score Gap Analysis. For the fine-tuned models and SaulLM, we also examine a Score Gap strategy. Instead of using a fixed score threshold, this method scans the ranked list from top to bottom and computes the score difference between adjacent candidates. Let s_i denote the score of the candidate at rank i. We define the local score gap as $\Delta_i = s_{i-1} - s_i$. Starting from the top-ranked candidate, we retain candidates until the first position where $\Delta_i \geq \gamma$, where γ is a run-specific gap threshold, selected separately for each run on the development set by grid search; this point is treated as a boundary between the leading score cluster and lower-confidence candidates. This makes the cut-off less rigid than a fixed threshold alone.

Forced Selection Mechanism. To avoid the extreme case in which no candidate survives filtering, we further apply a Forced-One Fallback. If the filtered set S_{final} is empty, the highest-scoring candidate is returned. This guarantees that each query receives at least one predicted noticed case and avoids a zero-recall outcome for that query.

Overall, this result selection procedure provides a practical way to convert ranking scores into final predictions. It combines rank-based pruning, score-based filtering, and a simple fallback mechanism to better handle variation in the number of relevant cases across queries.

5 Pipeline Development

The development of our system was an iterative process informed by comparative analysis on the COLIEE training data. In this section, we summarise the main alternatives we explored and explain how these observations informed the design of our final submission pipelines.

5.1 Alternative Methodology Evaluation

We first evaluated retrieval-only approaches, including BM25, dense retrieval, and a hybrid of these. While these methods achieved relatively high recall, they consistently suffered from low precision, as many retrieved cases were only topically related rather than legally relevant. This suggests that retrieving a large candidate set is insufficient without accurately identifying decision-supporting relationships.

We also tested representative re-ranking models from prior work, such as `BGE-Reranker-v2-m3`. While such models are effective on general-domain retrieval benchmarks, they were less competitive

in our legal retrieval setting. In particular, they appeared to favour topical similarity over deeper legal relevance, and therefore often failed to capture the case relationships needed for noticed-case identification. For this reason, we did not include these models in our final submission pipelines.

5.2 Selection of Final Pipelines

Based on the above observations, we conclude that a two-stage framework combining high-recall retrieval with strong re-ranking is necessary for this task.

We therefore adopt MonoT5 (both base and fine-tuned variants) and SaulLM-7B as our primary re-ranking models. These models provide complementary strengths: MonoT5 offers stable performance as a cross-encoder, while SaulLM-7B demonstrates enhanced capability in modelling complex legal reasoning. Instead of relying on a single architecture, we design three independent pipelines to explore different trade-offs between generalisation and domain-specific reasoning.

6 Experiments and Results

In this section, we present the evaluation of our proposed framework on the COLIEE 2026 Task 1 dataset.

6.1 Evaluation Metrics and Setup

Following the official COLIEE evaluation protocol, we report Precision, Recall, and F1-score. For the official submissions, our models were trained on the released COLIEE training set and then used to predict noticed cases for the unlabelled test set. No test labels were available at submission time. All experiments were performed on an Apple M2 Max device with 32GB unified memory.

6.2 Results

Table 1 presents the official evaluation results of our three submitted runs. Among them, `ucdcs3`, which uses SaulLM as the re-ranking model, achieves the best F1-score. The differences among the three UCD-CS runs are relatively small, with all three producing comparable overall performance.

Overall, our system ranks in the middle tier of the shared task. These results indicate that the proposed framework provides a competitive baseline, while also leaving clear room for further improvement.

6.3 Comparison of Submitted Runs

Although the three submitted runs share the same first-stage retrieval framework, they differ in the re-ranking model and final selection strategy. As discussed in Section 3.2, `ucdcs1` uses zero-shot MonoT5-Base as the re-ranker, `ucdcs2` uses a domain-finetuned MonoT5-Base model, and `ucdcs3` uses SaulLM-7B with logit-based scoring. As shown in Table 1, these three configurations achieve similar official F1-scores despite using different re-ranking models.

To further compare the behaviour of the three runs, we measured the overlap between their final predicted noticed-case sets using pairwise Jaccard similarity. For each query, we computed the Jaccard similarity between the returned case sets of each pair of runs and then averaged the scores over all queries. To distinguish overall output overlap from overlap in relevant retrieved cases, we also computed a correctness-restricted Jaccard similarity using only correctly returned noticed cases, i.e., after intersecting each run's prediction set with the gold noticed-case set for that query. The results of these are shown in Table 2.

Team	Run	F1	P	R
NOWJ	submission_2	0.4220	0.4235	0.4206
JNLP	random_forest	0.4126	0.4341	0.3931
SIL	submission_sil2	0.3871	0.4186	0.3600
mezza	task_1_mezzanino	0.3289	0.3161	0.3429
INTIT	3.intit_bm25_year	0.3165	0.3249	0.3086
DU	du3	0.3141	0.2945	0.3366
FLNLP	task1_flnlpltr	0.2935	0.2619	0.3337
UA	ua_run3	0.2666	0.2038	0.3851
UCD-CS	**ucdcs3**	**0.2645**	**0.2480**	**0.2834**
UCD-CS	**ucdcs1**	**0.2607**	**0.2131**	**0.3354**
UCD-CS	**ucdcs2**	**0.2565**	**0.2405**	**0.2749**
JUNLLP	task1_run2_results	0.2525	0.2193	0.2977
74688	task-1-ualbany	0.2346	0.2130	0.2611
UB2026	run1	0.2233	0.2338	0.2137
Sach	sach_task1_run2	0.2231	0.1631	0.3531
BJPWH	bjpwh3	0.2101	0.1510	0.3451
ABAI	run2recall	0.1771	0.1596	0.1989
AIIRLab	task1_aiirqwen	0.1516	0.2642	0.1063
bosch	bosch_task1_run	0.1466	0.0892	0.4103
KeioAndrewShin	task1_submission_v9	0.1383	0.1527	0.1263

Table 1: COLIEE Task 1 results ranked by F1 score. For each team, only the best-performing run (by F1) is reported, except for UCD-CS where all runs are included (P=Precision, R=Recall).

Run Pair	Overall Jaccard	Correct-only Jaccard
ucdcs1 vs ucdcs2	0.4741	0.7860
ucdcs1 vs ucdcs3	0.4864	0.8010
ucdcs2 vs ucdcs3	0.7137	0.8659

Table 2: Average pairwise Jaccard similarity between the final returned noticed-case sets of our three submitted runs, and between their correctly returned noticed cases only, computed over all test queries.

The results show that `ucdcs2` and `ucdcs3` are substantially more similar to each other than either is to `ucdcs1` in their overall predictions. However, the correctness-restricted Jaccard scores are markedly higher for all three run pairs, ranging from 0.7860 to 0.8659. This indicates that, although the runs often return different final case sets overall, there is a much stronger overlap for relevant noticed cases they retrieve. In other words, much of the disagreement between the submitted runs arises from differences in non-relevant returned cases rather than from retrieving fundamentally different relevant cases.

As an additional analysis, we compared the three submitted runs using ranking-oriented metrics and also evaluated an exploratory fusion of the three runs using Reciprocal Rank Fusion (RRF).

While the official submission results in Table 1 are reported in terms the set-based metrics of precision, recall, and F1, we further computed Mean Average Precision (MAP) in a *post-hoc* analysis

Run	MAP	F1	Precision	Recall
ucdcs1	0.2928	0.2607	0.2131	0.3354
ucdcs2	0.2427	0.2565	0.2405	0.2749
ucdcs3	0.2660	0.2645	0.2480	0.2834
RRF-fused (Top-5)	0.2699	0.2587	0.2460	0.3511

Table 3: Comparison of the three submitted runs and an exploratory RRF fusion. The three submitted runs are evaluated over their full returned lists, while the RRF-fused run is evaluated at top 5.

using the official ground-truth labels released by the COLIEE organisers. As shown in Table 3, ucdcs1 achieved the highest MAP (0.2928) amongst our runs, followed by ucdcs3 and then ucdcs2. This suggests that, although ucdcs3 performed best under the official F1-based evaluation, ucdcs1 produced the strongest ranking quality under this additional ranking-based analysis.

The Jaccard similarities shown above in Table 2 also motivated us to conduct an additional *post-hoc* analysis, on the basis that ranked lists with a high level of agreements on relevant documents can often be effectively aggregated under the "chorus effect" of data fusion [22]. We therefore fused the three submitted runs, again using RRF with the standard constant $K = 60$, and evaluated the fused ranking at top 5. This fixed cutoff was chosen as it is reflective of the typical cutoff values selected for the individual runs during training under the process outlined in Section 4.4 (which in practice ranged between 5 and 7 when optimising for F1). The fused run achieved MAP@5 of 0.2699, Precision@5 of 0.2460, Recall@5 of 0.3511, and F1@5 of 0.2587, as also shown in Table 3. While this setting does not exactly replicate the selection mechanisms used in the individual runs, it still allows for a broadly comparable analysis of ranking behaviour. Taken together with the Jaccard analysis, these results suggest that the three runs differ noticeably in their overall returned sets, but already overlap strongly in the correctly retrieved noticed cases, which limits the gains obtainable from simple post-hoc fusion.

6.4 Impact of Retrieval and Re-ranking

To analyse the role of each stage, we examine the performance of retrieval-only methods versus our best-performing full pipeline, which uses SaulLM for re-ranking. This comparison is based on our own post-hoc evaluation using the ground-truth labels released by COLIEE, and is therefore intended as an internal analysis of system behaviour rather than as part of the official competition results.

Method	MAP	F1	Precision	Recall
BM25 (Top-200)	0.3530	0.0634	0.0339	0.7866
Dense (Top-200)	0.3055	0.0348	0.0180	0.8210
Hybrid (Top-200)	0.3455	0.0359	0.0186	**0.8493**
Hybrid (Top-20)	0.3243	0.1784	0.1156	0.5760
Full Pipeline (SaulLM)	0.2660	**0.2645**	0.2480	0.2834

Table 4: Comparison between retrieval-only methods and the full pipeline.

As shown in Table 4, retrieval-only methods serve as an initial coarse filtering stage: they achieve high recall while substantially reducing the number of candidate cases that need to be examined. Their MAP values also show that they can rank some relevant cases near the top of the candidate list. However, their precision is extremely low, resulting in poor F1-scores. Applying neural re-ranking on these filtered candidates substantially improves precision and overall F1 performance, at the cost of recall and consequently MAP.

This result demonstrates that in legal case retrieval, retrieval acts primarily to narrow down the search space and ensure relevant cases are included, while re-ranking is essential for accurately identifying the most relevant cases.

7 Discussion

The Precision-Recall Trade-off in Legal Retrieval. The results in Table 4 show clear performance shifts across different stages of the pipeline. The hybrid retrieval stage (BM25 + BGE-M3) achieves high recall (0.8493), but recall decreases to 0.2834 in the full pipeline, together with a substantial increase in precision. This trade-off reflects the design and current limitation of our system. The first-stage retriever is intended to preserve a broad set of potentially relevant cases, and the top-200 candidate pool already contains many true noticed cases. However, the re-ranking stage does not consistently assign sufficiently high scores to all of these true noticed cases. As a result, some relevant cases remain in the candidate pool but are ranked below the final output cut-off and are not included in the final predictions.

In COLIEE Task 1, where the number of noticed cases varies across queries and is unknown at inference time, this ranking difficulty becomes especially important. Legal relevance is not determined by topical similarity alone, and some noticed cases may support the query case through more subtle reasoning links. This explains why the system achieves a large precision gain over retrieval-only baselines but does not obtain a proportional improvement in recall or overall F_1.

Beyond Semantic Similarity: Judicial Logic vs. Topicality. One important observation from our development experiments is the difference between general semantic similarity and legal relevance. General-purpose rerankers often assign high scores to case pairs that are topically similar but do not have a genuine relationship in terms of supporting legal reasoning.

Interestingly, the zero-shot MonoT5-Base model (ucdcs1) provided a strong baseline and outperformed our fine-tuned variant (ucdcs2). One possible explanation is that the available legal training data are limited, which may make the fine-tuned model more vulnerable to overfitting. At the same time, the better performance of SaulLM-7B suggests that legal-domain pre-training remains useful for capturing case relationships that depend on judicial reasoning, which is consistent with our previous findings on domain pre-training in the context of legal argumentation mining [25]. Our logit-based scoring method also allows SaulLM outputs to be used as continuous ranking scores rather than simple binary decisions.

Efficacy of Structural Legal Abstraction. The transition from full-text processing to structured *Legal Overviews* appears to play an important role. By distilling legal documents into structured components—[ANCHORS], [FACTS], and [LOGIC], we aim to reduce procedural noise and preserve the information most relevant to

noticed-case retrieval. This design is also motivated by the 'lost in the middle' phenomenon common in long-context Transformer architectures, where models tend to make less effective use of information located in the middle of long input sequences [15]. Similar structured prompting methodologies have also shown promise in handling extensive legal datasets where procedural noise often masks critical signals [10].

Limitations and Future Work. Our experimental results demonstrate that the hybrid retrieval stage achieves high recall (i.e., 0.8493 at $K = 200$), indicating that our framework provides a reliable candidate pool for downstream re-ranking. However, although the subsequent re-ranking stage improves precision, it does not produce a comparable gain in overall F_1, partly because the increase in precision is accompanied by a reduction in recall. This suggests that the current neural re-rankers may still struggle to consistently identify all true noticed cases from the top-200 candidates, especially when the relevant cases depend on complex logical signals rather than strong topical similarity.

Furthermore, the system remains sensitive to the quality of the LLM-generated overviews; missing critical legal anchors during the abstraction phase can lead to cascading errors in the downstream process. Despite the long-context capabilities of BGE-M3, our system also faces challenges in handling subtle, cross-case analogies that necessitate multi-hop reasoning.

Consequently, our future research will prioritise the exploration of more effective re-ranking architectures and specialised fine-tuning strategies specifically tailored for judicial logic. By investigating alternative model variants and more intensive domain-specific training, we expect to further bridge the gap between candidate retrieval and precise case identification.

8 Conclusion

In this work, we presented a multi-stage retrieval and re-ranking framework for legal case retrieval in COLIEE 2026 Task 1. The framework combines structured legal abstraction, hybrid first-stage retrieval, and neural re-ranking to address several core challenges of the task, including long judicial documents, vocabulary mismatch, and the difficulty of identifying decision-supporting case relationships.

Our experiments show that hybrid retrieval can provide a high-recall candidate pool, while downstream re-ranking improves the precision of the final predictions. The results also suggest that legal-domain adaptation remains important for this task, although accurately distinguishing subtle decision-supporting relationships between topically similar cases is still difficult. In particular, the gap between candidate recall and final ranking performance indicates that re-ranking remains the main challenge in the current system.

Future work will focus on stronger legal re-ranking strategies, more effective domain-specific training, and improved use of structured legal representations. These directions may help narrow the gap between broad candidate retrieval and precise noticed-case identification.

References

[1] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. https://arxiv.org/abs/1611.09268v3

[2] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets straight out of Law School. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 2898–2904. doi:10.18653/v1/2020.findings-emnlp.261

[3] Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 2318–2335. doi:10.18653/v1/2024.findings-acl.137

[4] Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre F. T. Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, and Michael Desa. 2024. SaulLM-7B: A pioneering Large Language Model for Law. doi:10.48550/arXiv.2403.03883 arXiv:2403.03883 [cs].

[5] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, Boston MA USA, 758–759. doi:10.1145/1571941.1572114

[6] Damian Curran and Mike Conway. 2024. Similarity Ranking of Case Law Using Propositions as Features. In *New Frontiers in Artificial Intelligence*, Toyotaro Suzumura and Mayumi Bono (Eds.). Springer Nature, Singapore, 156–166. doi:10.1007/978-981-97-3076-6_11

[7] Aniket Deroy, Kripabandhu Ghosh, and Saptarshi Ghosh. 2025. Applicability of large language models and generative models for legal case judgement summarization. *Artificial Intelligence and Law* 33, 4 (Dec. 2025), 1007–1050. doi:10.1007/s10506-024-09411-z

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. doi:10.18653/v1/N19-1423

[9] Yi Feng, Chuanyi Li, and Vincent Ng. 2024. Legal Case Retrieval: A Survey of the State of the Art. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 6472–6485. doi:10.18653/v1/2024.acl-long.350

[10] Wei Gao, Shuai Yu, Yongbin Qin, Caiwei Yang, Ruizhang Huang, Yanping Chen, and Chuan Lin. 2025. LSDK-LegalSum: improving legal judgment summarization using logical structure and domain knowledge. *Journal of King Saud University Computer and Information Sciences* 37, 1 (March 2025), 3. doi:10.1007/s44443-025-00022-5

[11] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Juliano Rabelo, Ken Satoh, and Masaharu Yoshioka. 2024. Overview and Discussion of the Competition on Legal Information, Extraction/Entailment (COLIEE) 2023. *The Review of Socionetwork Strategies* 18, 1 (April 2024), 27–47. doi:10.1007/s12626-023-00152-0

[12] Yoshinobu Kano, Randy Goebel, Mi-Young Kim, Calum Kwan, Ken Satoh, Hiroaki Yamada, and Masaharu Yoshioka. 2026. An Overview of the COLIEE 2026 Competition: Legal Case Law and Statute Law Information Retrieval and Entailment. In *Proceedings of the 21st International Conference on Artificial Intelligence and Law (ICAIL 2026)*.

[13] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, 9459–9474. https://dl.acm.org/doi/10.5555/3495724.3496517

[14] David Lillis. 2020. On the Evaluation of Data Fusion for Information Retrieval. In *Forum for Information Retrieval Evaluation (FIRE '20)*. Hyderabad, India. doi:10.1145/3441501.3441506

[15] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics* 12 (2024), 157–173. doi:10.1162/tacl_a_00638

[16] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. doi:10.48550/arXiv.1907.11692 arXiv:1907.11692 [cs].

[17] Hoang-Trung Nguyen, Tan-Minh Nguyen, Xuan-Bach Le, Tuan-Kiet Le, Khanh-Huyen Nguyen, Ha-Thanh Nguyen, Thi-Hai-Yen Vuong, and Le-Minh Nguyen. 2025. NOWJ@COLIEE 2025: A Multi-stage Framework Integrating Embedding Models and Large Language Models for Legal Retrieval and Entailment. In *Proceedings of the Workshop on the Twelfth International Competition on Legal*

Information Extraction and Entailment (COLIEE 2025). Chicago, USA, 47–56.

[18] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 708–718. doi:10.18653/v1/2020.findings-emnlp.63

[19] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 3980–3990. doi:10.18653/v1/D19-1410

[20] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (April 2009), 333–389. doi:10.1561/1500000019

[21] Olga Shulayeva, Advaith Siddharthan, and Adam Wyner. 2017. Recognizing cited facts and principles in legal judgements. *Artificial Intelligence and Law* 25, 1 (March 2017), 107–126. doi:10.1007/s10506-017-9197-6

[22] Christopher C Vogt and Garrison W Cottrell. 1999. Fusion via a linear combination of scores. *Information Retrieval* 1, 3 (1999), 151–173. doi:10.1023/A:1009980820262

[23] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, 24824–24837.

[24] Deiby Wu, Sarah Lawrence, and Behrooz Mansouri. 2025. AIIR Lab at COLIEE 2025: Exploring Applications of Large Language Models for Legal Text Retrieval and Entailment. In *Proceedings of the Workshop on the Twelfth International Competition on Legal Information Extraction and Entailment (COLIEE 2025)*. Chicago, USA, 112–118.

[25] Gechuan Zhang, David Lillis, and Paul Nulty. 2021. Can Domain Pre-training Help Interdisciplinary Researchers from Data Annotation Poverty? A Case Study of Legal Argument Mining with BERT-based Transformers. In *Proceedings of the Workshop on Natural Language Processing for Digital Humanities (NLP4DH)*. Association for Computational Linguistics, 121–130. https://aclanthology.org/2021.nlp4dh-1.14

[26] Gechuan Zhang, Paul Nulty, and David Lillis. 2023. Argument Mining with Graph Representation Learning. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law (ICAIL '23)*. Association for Computing Machinery, New York, NY, USA, 371–380. doi:10.1145/3594536.3595152

[27] Yucheng Zhou, Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Guodong Long, Binxing Jiao, and Daxin Jiang. 2023. Towards Robust Ranker for Text Retrieval. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 5387–5401. doi:10.18653/v1/2023.findings-acl.332

A Prompt for Legal Semantic Abstraction

For transparency and reproducibility, we provide below the exact prompt used to generate the structured case overviews in our implementation.

```
You are an expert Legal Case Analyst. Your goal is to extract key
information from the provided Canadian Federal Court judgment to help
find similar cases in a database.

**INSTRUCTIONS:**
Analyze the text and extract information into the following THREE sections.

1. **[ANCHORS]** (High Priority):
  - **KEYWORDS**: Extract or generate 5-7 distinct legal terms
    (e.g., "Family Class", "Adoption", "Procedural Fairness").
    *Crucial: If the case turns on the interpretation of a foreign
    country's law, MUST include the tag "Foreign Law Interpretation".*
  - **STATUTES**: List EXACT statute names and section numbers cited.

2. **[LOGIC]** (High Priority):
  - **Issue**: What is the primary legal question?
  - **Ratio**: The core reasoning for the decision.
    *Focus on the specific legal error or principle.
    Do NOT output generic "Standard of Review" boilerplate.*

3. **[FACTS]** (Medium Priority):
  - Provide a concise summary (max 50 words) of the factual background.
    Who are the parties? What triggered the dispute?

**CONSTRAINTS:**
- Output format must use the exact headers: [ANCHORS], [LOGIC], [FACTS].
- **CRITICAL**: You MUST write "[END]" immediately after the Facts
  section to signal completion.

**TEXT TO PROCESS:**
'''
{text}
'''

**FINAL CHECKLIST FOR YOU:**
1. Did you include [ANCHORS]?
2. Did you include [LOGIC]?
3. Did you include [FACTS]? (DO NOT FORGET THIS SECTION)
4. Did you end with [END]?

**OUTPUT:**
```